Maintaining Privacy and Utility with Custom Loss Functions in Location-Based Applications

[1]Gagandeep Kaur
[1]Research Scholar
Department of Computer Science Engineering
Chandigarh University
Mohali, Punjab, India -144601

Email: gaganarora.cu19@gmail.com

[2]Ruchika Gupta
[2]Professor
Department of Computer Science Engineering
Chandigarh University
Mohali, Punjab, India -144601
rgupt0009@gmail.com

***Abstract*—** In the course of this research, a bespoke function was developed to strike a balance between privacy preservation and data utility. This function is specifically designed to ensure the usability of location-based data while simultaneously safeguarding user privacy. The function employs a gradient descent optimization process to identify the optimal transformation for the dataset that is to be perturbed. This is a crucial step as the original data, if left unaltered, could be vulnerable to privacy breaches. To achieve the perturbation, Laplacian noise was introduced into the original data, with the custom function serving as the objective function for the transformation of the original location based dataset the effectiveness of this perturbation operation was then evaluated using a suite of 26 machine learning algorithms. This was premised on the assumption that potential attackers might utilize machine learning models to infringe upon user privacy. The evaluation was based on a variety of metrics, leading to the conclusion that the methodologies employed in this research were successful. This conclusion is substantiated by a series of rigorous experiments and a detailed case study of homogeneity attack conducted as part of this research.

Keywords: Privacy, Utility of data, Machine Learning, Location based Services.

I. INTRODUCTION

Predictability of the data depends on multiple factors and if a hacker understands the factors based on which the prediction can be made he/she can compromise the systems and bring security challenges in terms of privacy maintenance [1]. The contemporary literature gives information on a large number of factors based on which the anti-social elements are able to exploit and compromise the systems. The reasons include usage of weak passwords, not following proper security protocols; weak security architecture and many other factors contribute to the failure of system security. Besides this, copious evidence can be found in the databases of incidence reports of breach of privacy and compromises of

security infrastructure of the organizations [2] [3] that work to safeguard the privacy of the people at large .

The most critical component of the security breach is access to the data and sophistication of the technologies used to compromise privacy. Parallelly, governments , corporations and individuals whose job is to protect have developed "privacy enabling technologies" or P.E.T. Multiple descriptive statistical analysis of reports of incidents [3] of privacy breach reveals that the most frequent reasons for breach are insider attack, not updating the software , physical theft , malware etc. However in the case of the mobile geodata the breach is done using more sophisticated technological weaponry such as use of machine learning, deep learning, statistical analysis etc., that helps to make educated guess on location of the target, In each country there are multiple laws that try to deter the people who take advantage of the gaps in the securities and fortification of digital assets. However, the implementation of such laws varies from one country to another and the most reliable solution basically lives with highly technologically equipped organizations [4] and with those who deal with the ethical issues [4] and dilemmas [5] that arise from the implementation of location based data sharing.
Technologies that are based on cryptographic algorithms are utilized the vast majority of the time in order to preserve the users' and digital assets' privacy [6]. Holomorphic encryption algorithms have consistently demonstrated their worth throughout history. In particular, techniques such as fully homomorphic encryption, differential privacy, and zero-knowledge proofs have had a considerable influence on the information security sector [7]. Researchers have been integrating several encryption sources in order to better strengthen these cryptographic algorithms [8] Obfuscation [9], Pseudonymization [10] [19], data minimization, communication anonymizer [11], encryption, differential privacy [20], k-anonymity [21], l-diversity [22], and t-closeness [23] are some of the methods and techniques employed for masking and security hardening. These are frequently utilized to maintain privacy, especially in location-based applications where user location data is particularly sensitive. Additionally, combinatorial methods that use multiple permutations of these algorithms, as well as more advanced techniques like homomorphic encryption [24] and secure multi-party computation [25], are being explored for their potential to provide even stronger privacy guarantees. In the context of location-based services, techniques such as geo-in distinguishability [26] and spatial cloaking [27] have been specifically designed to protect user location privacy.

The structure of this paper is as follows. The second section summarizes information and analysis from contemporary relevant work. The third section discusses the formulation of the problem. The construction of the proposed privacy maintenance algorithm is discussed in Section 4. Design and implementation algorithm are conducted in Section 5. Section 6 find out the results and in Section 7 concludes the study and provides recommendations for the future based on its findings.

II. LITERATURE REVIEW

Looking at recent studies, we can see that a lot of work has been done to create ways to keep location-based apps private. There are many examples of methods and systems used to keep privacy safe. One of these methods is called differential privacy. This is a way of keeping information private by adding random noise, or extra data, to the original location data before it is processed or analysed. This method is often used in machine learning and data mining, where it has been successful in keeping people's information private while still making the data useful. Recently, there has also been an increase in the use of secure multi-party computation (MPC) techniques. These techniques allow multiple parties to work together to compute a function on their private data without revealing any information about their data. MPC is used in areas like financial modeling, genomic analysis, and social network analysis. In these

fields, it's important to analyse sensitive data while keeping people's information private. According to many recent surveys and reviews [9] [10] [11] [13], advances in cryptography have enabled the development of holomorphic encryption, which allows data to be encrypted while retaining their computational utility. This has the potential to revolutionize data processing and analysis because it eliminates the need to decrypt data before using it. Holomorphic encryption has been utilized in a number of fields, including healthcare, finance, and e-commerce, where sensitive data must be analysed while maintaining the privacy of individuals. There have also been significant advancements in the field of data publication that protects privacy. This involves releasing data in a manner that protects the privacy of individuals while retaining the data's analytic utility. A common method is k-anonymity, which involves replacing individual identifiers with group identifiers so that no individual can be identified from the data [21]. Other approaches include i-diversity, t-closeness, and methods based on differential privacy [22]. Each method has its pros and cons; however, the key factor that plays a role in promoting the vulnerability is the inherent nature of the Location based data itself. Location-based information is among the most sensitive types of personal information. It reveals not only the locations of individuals, but also their routines, interests, and habits. Consequently, it is essential to protect the privacy of location-based data. Consequently, the characteristics of the data itself present significant obstacles to achieving this objective of remaining safe in terms of privacy.

Clearly, the location-based data is spatial-temporal by nature. It includes information regarding the location and time of individuals. This data's spatiotemporal nature generates a high-dimensional space, making it difficult to anonymize or obfuscate it without significantly diminishing its utility [12] [14]. Any effort to alter the data in order to safeguard privacy must also preserve its spatiotemporal coherence. Second, location-based data is frequently shared by app developers, third-party advertisers, and service providers. This data sharing creates a complex data ecosystem, making it difficult to enforce privacy policies and regulations. In addition, each entity may have different data retention policies and data sharing agreements, resulting in a data landscape that is difficult to manage. Moreover, location-based data can expose sensitive information about individuals, such as their political beliefs, religious convictions, activities or health conditions. By analyzing patterns in the data, such as frequent visits to specific locations or changes in mobility behavior, this sensitive information can be inferred. As they do not necessarily violate any explicit privacy policies or regulations, such inference attacks can be difficult to detect and prevent.

Location-based data are susceptible to a variety of attacks, including inference attacks, de-anonymization attacks, and linkage attacks. These attacks exploit data vulnerabilities, such as correlations between location data and external datasets, in order to disclose the identity of individuals or sensitive information. Protecting against these attacks requires a comprehensive privacy framework that takes into account the different types of attacks and the data's properties. As a result of the spatiotemporal nature, complex data ecosystem, sensitivity, and susceptibility to a variety of types of attacks, location-based data pose significant challenges to privacy protection. Defeating these obstacles requires a comprehensive strategy that takes into account the diverse properties of the data, the data ecosystem, and the potential attack scenarios.

a. Inferences From Literature Review

From the literature, it's evident that the vulnerability of location-based applications is influenced by several factors, including the degree of randomness and similarity of patterns that can be discovered in the data. The entropy of the data, denoted as E (D) [25], which can be interpreted as the degree of randomness in the information, is one of these factors. For example, a dataset 'D' with low entropy might contain many repeated values, making it easier for adversaries, denoted as 'Adv', to deduce the original data. When the

data exhibits a high level of homogeneity, denoted as H(D), it indicates a low level of variability. This makes it easier for attackers to decipher the data in its original form. An example would be a dataset that contains a single location repeated numerous times, simplifying the task for an adversary to determine the initial location. Such mechanisms of guessing the user's attributes are known as homogeneity attacks [22] [23] [24].If the data is highly skewed, denoted as Sk(D) [26] [27], it suggests a disproportionately high number of values falling on one side of the distribution compared to the other. For instance, if the majority of the data is concentrated in a small number of locations, it becomes much easier for an adversary to determine the original location of the data. The existence of a pattern in the location data, denoted as P(D), can be inferred from the fact that the data exhibits a high degree of regularity. If a user, denoted as U, consistently visits the same location at the same time every day, for instance, it becomes easier for an adversary to deduce the identity and location of the user.

Human behavior, denoted as B(U), is another factor to consider when evaluating the risk associated with location-based applications. If a user, U, frequently visits the same location, such as their workplace or home, an adversary, Adv, can more easily determine the user's identity and location with the help of queries, Q [17]. Another contributing factor to the vulnerability of location-based applications is the proximity of successive locations, denoted as P(L). If the user's location data shows a high proximity between successive locations, it suggests that the user is likely staying in the same place for a significant amount of time. This makes it easier for attackers to determine the user's location [17][20]. Therefore, it's crucial to consider these factors when developing location-based applications to ensure adequate protection of users' privacy. In this research work, the aim is to construct a framework to address this challenge, leveraging the advancements in new machine learning algorithms [30] . Several measures can be used to describe the shape of a distribution, with the most common ones being Skewness, denoted as Sk(D), homogeneity, H(D), and kurtosis, K(D) [28]. Skewness measures the asymmetry of a distribution, homogeneity measures the similarity of values within a distribution, and kurtosis measures the peakedness or flatness of a distribution. The relationship between Skewness, kurtosis, and homogeneity can vary depending on the data distribution. For a normal distribution, for instance, the Skewness is zero, the kurtosis is three, and the homogeneity is also three. However, for other distributions, such as a bimodal distribution or a highly skewed distribution, these values will differ. High skewness or bimodality may indicate a higher vulnerability to attacks on location privacy. Maintaining privacy and utility, U [12], in location-based apps is a challenging problem that requires careful consideration of both technical and ethical issues. Custom loss functions, L [12] [14], are a promising approach to address the privacy-utility trade-off in location-based apps. By leveraging the power of machine learning, we can build location-based apps that provide useful services to users while protecting their privacy [15]. In the next section, we will formulate the problem, denoted as P, so that a reliable solution can be formulated and addressed.

III. FORMULATION OF THE PROBLEM

After a comprehensive review of relevant research, it's evident that the inherent characteristics of the location dataset, denoted as D, pose significant challenges to protecting individuals' privacy in location-based applications. The generally homogeneous nature of the location dataset makes it easy for adversaries, denoted as Adv, to guess the target's location. Moreover, recent advancements in machine learning and deep learning algorithms have enabled multiple adversaries to triangulate the target's location, denoted as (x, y). Therefore, the vulnerability of location-based applications can be attributed to a variety of factors, including entropy (e), homogeneity (h), Skewness (s), regularity (r), proximity of successive locations (p), and human behaviour (b). The location dataset is represented by a matrix D, where D[i,j] denotes the j-th attribute value for the i-th location. The set of all possible locations that a target individual can be present in is represented by E. The homogeneity of the location dataset, which makes it easy for

adversaries to guess the target's location, poses a problem for maintaining the privacy of the target individual's location. Advancements in machine learning and deep learning algorithms have made it possible for multiple adversaries to triangulate the target's location (x, y).

Factors such as entropy (e), homogeneity (h), Skewness (s), regularity (r), proximity of successive locations (p), and human behavior (b) contribute to the vulnerability of location-based applications. To maintain privacy, it's necessary to periodically check these parameters. If the values of these parameters individually decrease or the data becomes more prone to attack, the value of the metric 'proneness' (pn) also decreases. Once the proneness metric (pn) drops below a certain threshold, randomness and perturbation can be introduced into the dataset D. The goal of introducing randomness and perturbation is to prevent the target individual's location and its attributes from being easily guessed or triangulated by adversaries, while still preserving the utility of the data for legitimate purposes.

a. SOLUTION

Let's denote the location dataset as a matrix D, where D[i,j] represents the j-th attribute value for the i-th location. Let E be the set of all possible locations that a target individual, denoted as T, can be present in.
Let X be a random variable representing the location of the target individual T, such that X takes values in E. Let Y be a random variable representing the attributes of the target individual T at their location, such that Y takes values in the set A of all possible attribute values for the dataset D.
Let Z be a random variable representing the perturbed version of the dataset D, such that Z[i,j] denotes the j-th attribute value for the i-th location in the perturbed dataset.

To maintain both the privacy and the utility of the dataset D, we can introduce a privacy loss function, denoted as L_p(), into the learning objective of the machine learning algorithm. This function measures the degree of privacy loss, denoted as PL, and simultaneously the utility loss, denoted as UL, resulting from the mapping M between the perturbed dataset Z and the target individual's location X and attributes Y.
A trade-off parameter, denoted as λ, is computed to balance the utility and privacy objectives of the machine learning algorithm. This parameter λ determines the relative importance of the privacy loss function L_p() compared to the utility objective, denoted as U, of the machine learning algorithm.
The resulting model, denoted as M* can then be used to predict the location X and attributes Y of a target individual T in a perturbed version of the dataset Z, while preserving their privacy.

IV. METHODOLOGY

In this section, the implementation of the solution will be explained and demonstrated using real life examples and cases, based on the problem that was formulated and the purpose solution that was intended. For better understanding of the research flow followed it is requested to the reader to check the block diagram Fig. [1].
The dataset D Fig. [2] with mxn dimensions is prone to the homogeneity attack, due to factors such as entropy ‘e’, homogeneity ‘h’, Skewness ‘s’, regularity ‘r’, proximity of successive locations ‘p’ and other attributes such a location [x,y] form the location dataset of a user 'U'.

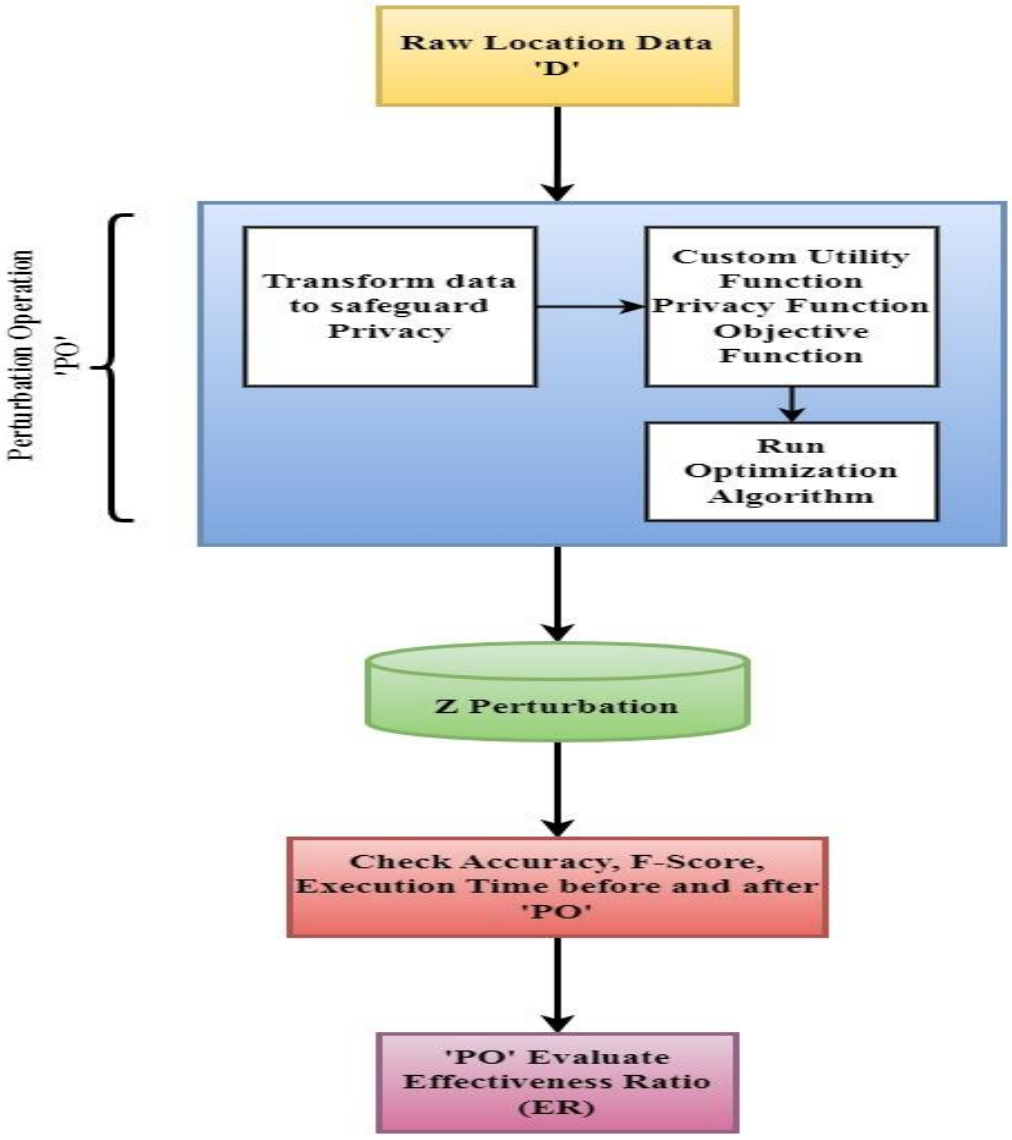


Fig 1. Block Diagram

	user id	day	activity	location	lat	lon
0	1	1	sleeping	home	37.685531	-121.805087
1	1	2	sitting	park	37.964524	-122.432272
2	1	2	sitting	road	37.646569	-121.898505
3	1	3	sleeping	home	37.696436	-121.896857
4	1	3	sitting	road	37.690617	-121.852963
..	...	...	...	...	...	...
106	2	6	sleeping	home	37.693749	-121.897400
107	2	6	sitting	mall	37.806811	-122.251755
108	2	6	sleeping	mall	37.892208	-122.275577
109	2	7	running	office	37.710774	-122.086909
110	2	7	walk	road	37.658312	-121.814864

Fig 2. Location dataset 'D'

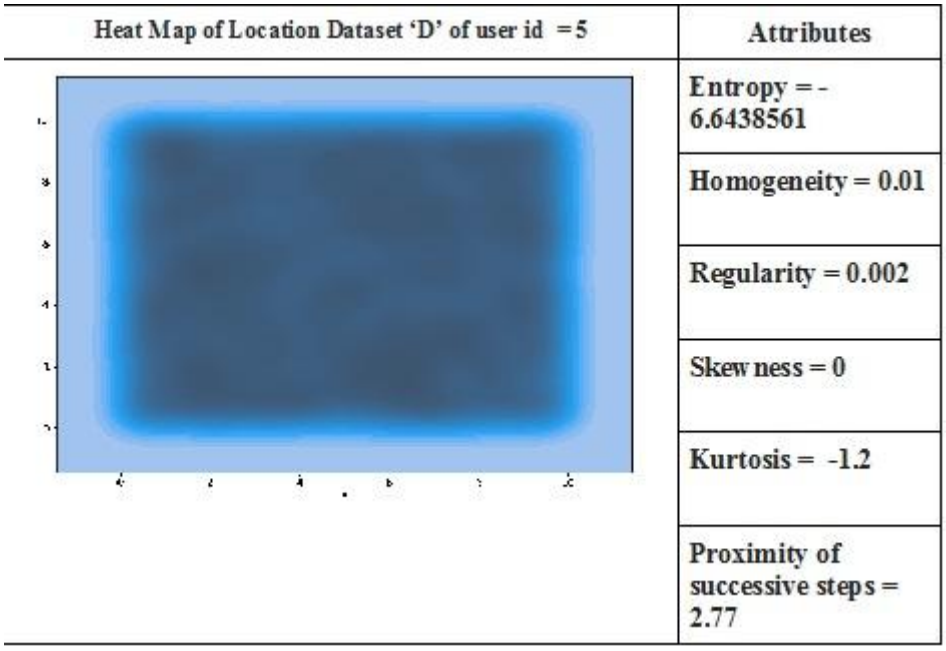


Fig 3. Heat map and the attributes the original dataset 'D'

Entropy, denoted as E, is a well-established measure of the randomness or uncertainty of location data. A greater negative value, for instance -6,643, indicates that the data is less unpredictable and less random, making it easier for an attacker, denoted as Adv, to identify individuals. However, the homogeneity values, denoted as H, in this instance are advantageous for users of location-based applications. A homogeneity value of 0.01 indicates that locations are extremely dissimilar, making it challenging for an attacker to identify individuals based on their location data Fig. [3].

Similar observations can be made for the metric Regularity, denoted as R. This metric measures the predictability or regularity of the location data. A value of 0.002 indicates that the locations are extremely irregular, making it challenging for an attacker to identify individuals based on their location data.

As its value is zero, the data shape is symmetric, which makes it more difficult for an attacker to identify individuals based on their location data. Similarly, the Kurtosis values, denoted as K, measure the peakiness of the location data distribution. A negative value indicates a distribution with fewer peaks than a normal distribution, making it harder for an attacker to identify individuals based on their location data.

However, in the case of Proximity to successive steps, denoted as P, the value (2.77) is not particularly high. An increase in value is desired. A greater value for this metric indicates that the data is more dispersed, making it harder for an attacker to identify individuals based on their location data. Here, the opposite is true. In this instance, it can be observed that the values of some of the factors are beneficial to the privacy of location-based app users, while others are not.

Consequently, employing a custom metric such as proneness of attack or "attack susceptibility", denoted as "pn", could facilitate the comprehension of the intensity of the attack vectors. With the use of the composite metric 'pn', interpretation will also be simplified. Hence, the 'pn' can be calculated mathematically using a specific expression.

$$\text{Attack susceptibility 'pn'} = \max\{0, [(E * w1) + (H * w2) + (R * w3) + (S * w4) + (K * w5) + (P * w6)]\}$$

where w1, w2, w3, w4, w5, and w6 are the weights assigned to each metric, and max{0, x} returns x if x is non-negative and 0 otherwise.

'Using the values provided, and assuming equal weighting, the proneness of attack would be:

Proneness of Attack or pn = | (1/6 * -6.64) + (1/6 * 0.01) + (1/6 * 0.002) + (1/6 * 0) + (1/6 * -1.2) + (1/6 * 5.767411149346671) | = 1.059891564195784

The proneness of attack 'pn' metric was constructed using a naive method of computing because it is prone to bias because there is a larger volume and a high number of outliers. This method of computing was chosen because it is naive. Because of recent developments in machine learning, the adversaries may have an additional advantage to predict the location of the target. In light of this, the application of machine learning to the dataset will be carried out in this section with the intention of foiling the attackers' plans. Hence, ***mathematically***, Let D be the original location dataset represented by a matrix, where D[i,j] denotes the j-th attribute value for the i-th location. Let E be the set of all possible locations that a target individual can be present in. Let X be a random variable representing the location of the target individual, such that X takes values in E. Let Y be a random variable representing the attributes of the target individual at their location, such that Y takes values in the set of all possible attribute values for the dataset.

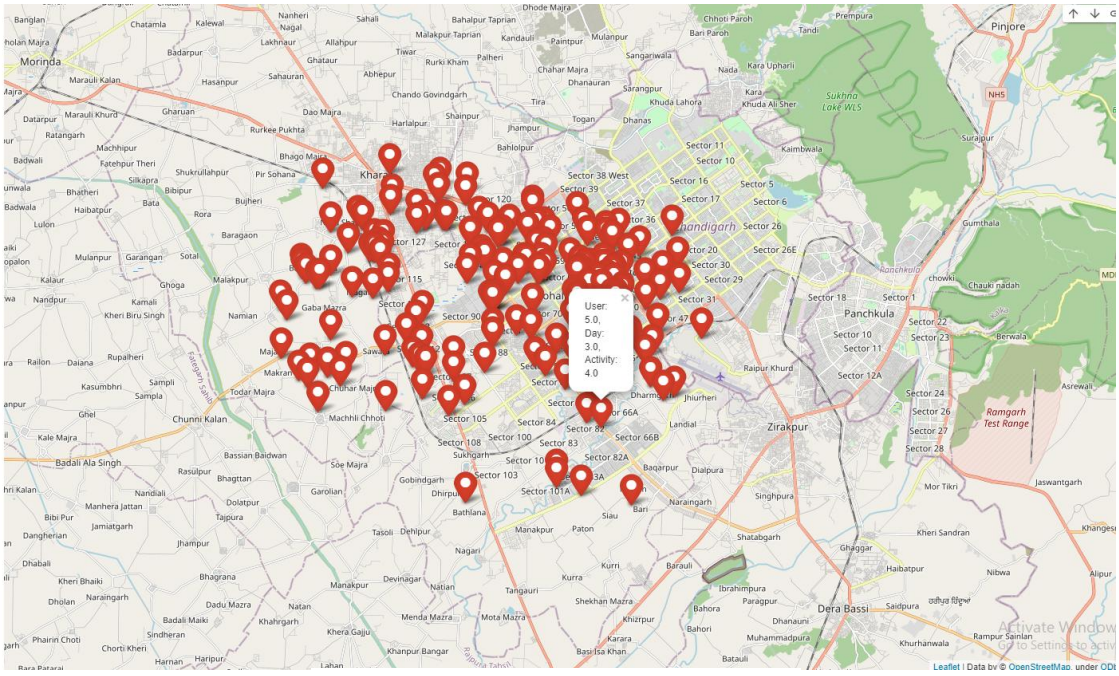
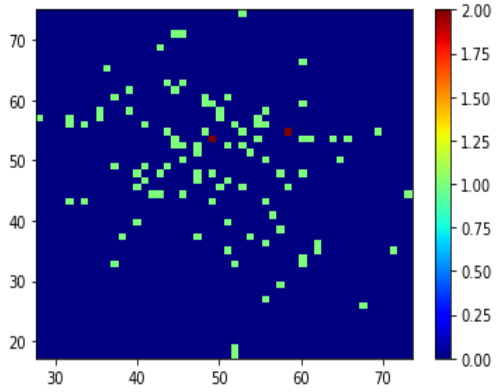


Fig 3. Visualisation of the Locations

Let Z be a random variable representing the perturbed version of the dataset D, such that Z[i,j] denotes the j-th attribute value for the i-th location in the perturbed dataset. For implementing this statement and step , a python function '**add_noise_to_perturb**' that takes as input the original dataset 'D' data and the privacy budget parameter epsilon was constructed Fig. [4]. The function calculates the sensitivity of the dataset, which is the maximum difference between two neighboring rows. It then calculates the scale of the Laplace distribution based on the sensitivity and privacy budget. Finally, it generates noise with Laplace distribution and adds it to the dataset. The perturbed dataset 'Z' is returned. Here is the output based on the equation

$Z = D + L(\varepsilon)$.

Location Data	**Attributes of Original Dataset 'D'**	
	**Entropy**	5.6842
	Homogeneity	0.9623
	Regularity	0.96236
	Skewness	-0.1941
	Kurtosis	0.24530
	Proximity of successive steps	21.932

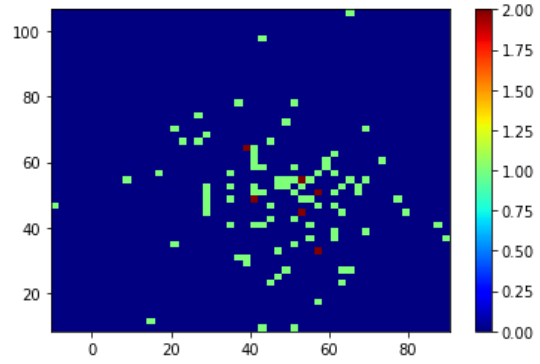
Perturbed dataset 'Z'	**Attributes of Dataset 'Z'**	
	**Entropy**	-inf
	Homogeneity	0.8952
	Regularity	0.8952
	Skewness	-0.122
	Kurtosis	1.0482
	Proximity of successive steps	35.742

Fig 4. Effect of adding noise to perturb on original dataset 'D'

From Figure 4, it is evident that the addition of Laplace noise, denoted as L(ε), has significantly altered the characteristics of the dataset, as indicated by the provided metrics. The entropy of the perturbed dataset, denoted as E_p, is significantly lower, indicating a reduction in the amount of information contained in the data. Both the homogeneity, denoted as H_p, and regularity metrics, denoted as R_p, have improved, suggesting that the data is now less consistent and more variable. The values for Skewness, denoted as S_p, and kurtosis, denoted as K_p, have also changed, suggesting a shift in the data distribution. Finally, there has been an increase in the proximity of successive steps, denoted as P_p, indicating that the data is less clustered and more dispersed. However, when it comes to the utility of the dataset, the perturbed data may become less useful. Especially, when the skewness S_p is close to zero, this indicates that the distribution of the location data is symmetric. This can simplify the task for the machine learning algorithm, denoted as ML, to recognise patterns in the data and make accurate predictions. Similarly, if the

kurtosis K_p is high (i.e., if the distribution has more peaks than a normal distribution), it can also simplify the task for the ML algorithm to recognise patterns and make accurate predictions, both of which are desirable from a utility perspective. The same is true in the case of homogeneity H_p and regularity R_p. Therefore, there is a limit to which the data can be transformed to safeguard privacy. Hence, in the next section, we explain the design of a privacy and utility loss function, denoted as L_pu(), to find a trade-off value of the privacy budget factor "epsilon", denoted as ε.

Hence, the original dataset D consists of user attributes and their corresponding locations, i.e.
D = { (x1, y1, z1, a1), (x2, y2, z2, a2), ..., (xn, yn, zn, an) }
Where xi, yi, zi are the coordinates of the user's location, and ai represents the user's attributes. We have a perturbed dataset Z | D' by adding Laplace noise to the location coordinates of D, i.e.
D' = { (x1', y1', z1', a1'), (x2', y2', z2', a2'), ..., (xn', yn', zn', an') }
Where xi' = xi + Li(ε), yi' = yi + Li(ε), and zi' = zi + Li(ε), and Li(ε) is the Laplace noise added to coordinate i with privacy budget ε., where ε=[0.1] .

Since, in our case of privacy-preserving data analysis, the purpose of applying machine learning algorithms is to make predictions or derive insights from the perturbed dataset **Z**, while at the same time protecting the privacy of individuals whose data is contained in the dataset. This is because since, since, in our case of privacy-preserving data analysis, the purpose of applying machine learning algorithms is to make predictions or derive insights from the perturbed dataset **Z,** Even when the model is trained on the perturbed dataset, the machine learning algorithms must be designed in such a way that they do not reveal sensitive information about individuals.
In this next section, we design a custom algorithm that consists of two loss functions to construct a perturbed dataset **Z.**

V. DESIGN AND IMPLEMENTATION

In this section, the design and implementation of the privacy loss and utility function is discussed. We take the help of the optimization approach (grad X) to find the perturbed dataset '**Z**', while varying the privacy budget (***epsilon***) and the value of alpha. Following are the assumptions, axioms and concepts behind constructing the custom algorithm to thwart homogeneity attacks.

a. ***Assumptions:***

- The default value of Alpha value is 0.5, implying equal importance to privacy and utility.

b. ***Axioms:***

- Privacy loss function takes original dataset D and perturbed dataset **Z** as input, and outputs a privacy loss score.
- Utility function takes original dataset D and perturbed dataset **Z** as input, and outputs a utility score.
- A balancing parameter alpha is defined to find the trade-off between privacy and utility.

c. *Concept/ Theory:*

- The objective of the optimization problem is to Find the perturbed dataset Z' that minimises P(D,Z) subject to the constraint U(D,Z) >= (1-alpha)*U(D,Z), where P is the privacy loss function and U is the utility function.
- Gradient descent optimization algorithm is used to solve the optimization problem and obtain the perturbed dataset **Z**.

Based on the assumptions and axioms mentioned above, we have developed an algorithm to find a perturbed dataset Z that minimizes the privacy loss function P(D, Z) subject to the constraint that the utility of Z is greater than or equal to (1-alpha) times the utility of D, where alpha is a parameter that balances the importance of privacy and utility.

Table 1. Utility and Custom Loss Optimization Algorithm

Step 0	**Variable Initializations**
	factors = ['Entropy', 'Homogeneity', 'Regularity', 'Skewness', 'Kurtosis', 'Proximity'] pthresholds = [0.7, 0.5, 0.8, 0.2, -0.2, 0.5] weights = [0.2, 0.2, 0.2, 0.2, 0.2, 0.1] Min_threshold = 3

Step 1	**def privacy_loss(D, Z)**:
	# Compute the factors for the perturbed dataset Z
	entropy = compute_privacy_entropy(Z)
	homogeneity = compute_privacy_homogeneity(Z)
	regularity = compute_privacy_regularity(Z)
	skewness = compute_privacy_skewness(Z)
	kurtosis = compute_privacy_kurtosis(Z)
	proximity = compute_normalised_privacy_proximity(Z)
	
	# Check if the factors meet the desired thresholds
	num_within_threshold = 0
	for factor, threshold in zip(factors, pthresholds):
	if eval(factor.lower()) >= threshold:
	num_within_threshold += 1
	
	# Compute the proneness of attack using a weighted average
	weights = [0.15, 0.15, 0.15, 0.15, 0.15, 0.1]
	proneness = np.dot(weights, factors)
	
	# Check if privacy is maintained
	if num_within_threshold >= min_threshold and proneness < 0.5:
	return 0
	else:
	return 1

Step 2	# Define the factors and their thresholds
	uthresholds = 1- pthresholds
	
	def utility_loss(D, Z):
	  # Compute the factors for the perturbed dataset Z
	  entropy = compute_utility_entropy(Z)
	  homogeneity = compute_utility_homogeneity(Z)
	  regularity = compute_utility_regularity(Z)
	  skewness = compute_utility_skewness(Z)
	  kurtosis = compute_utility_kurtosis(Z)
	  proximity = compute_normalised_utility_proximity(Z)
	
	  # Check if the factors meet the desired thresholds
	  num_within_threshold = 0
	  for factor, threshold in zip(factors, uthresholds):
	  if eval(factor.lower()) >= threshold:
	  num_within_threshold += 1
	
	  utility_loss = 1 - np.dot(factors)
	
	  # Check if utility is maintained
	  if num_within_threshold >= min_threshold and utility_loss < 0.5:
	  return 0
	  else:
	  return 1

Step 0	Initialise the perturbed dataset D' with the original dataset D. Compute the privacy loss function P(D, Z) and the utility function U(D, Z) for the current value of Z. Epsilon = [0,0.1,0.2…1]

Step 1	**define optimise_algorithm(D, alpha,epsilon)**:
	  # Define the initial perturbed dataset Z
	  D_perturbed = D
	
	  # Define the learning rate and number of iterations
	  learning_rate = 0.1
	  num_iterations = 100
	
	  # Define the privacy loss and utility loss functions
	  privacyFlag = privacy_loss(D, D_perturbed[**epsilon**]):
	  privacyFlag = utility_loss(D, D_perturbed[**epsilon**]):

```
# Run gradient descent optimization algorithm
for i in range(num_iterations):
    # Compute the gradients of the privacy loss and utility loss functions
    d_privacy_loss = grad(privacy_loss)(D, D_perturbed)
    d_utility_loss = grad(utility_loss)(D, D_perturbed)

    # Update the perturbed dataset Z
    D_perturbed -= learning_rate * d_privacy_loss / (1 - alpha) * d_utility_loss

return D_perturbed
```

The algorithm in Table 1. first Initialises the perturbed dataset Z with epsilon default value = 0.1 with the original dataset D. Then compute the privacy loss function P(D, D') and the utility function U(D, D') for the current value of D'. In the third step , we check if the current value of D' satisfies the utility constraint U(D, D') >= (1-alpha)U(D, D). Further ,we check If the utility constraint is not satisfied, run gradient descent optimization algorithm to update Z . The next step is to check for utility. If the utility constraint is satisfied, check if the privacy loss function P(D, Z) is minimised by checking if at least three factors meet the desired thresholds and the proneness of attack is below a certain threshold. Lastly , it is checked, If privacy is maintained, return the perturbed dataset Z. Otherwise, run the gradient descent optimization algorithm again with a new initialization of Z and repeat steps of checking alpha and epsilon until privacy and utility are both satisfied.

In a series of experiments, the value of alpha is varied and steps 4-6 are repeated to explore the privacy-utility trade-off space and determine an alpha value that achieves the desired level of privacy protection while maintaining sufficient utility for the intended use of the data. In the next section, the outcomes of these experiments are evaluated to ascertain the performance of custom privacy loss and utility functions.

VI. RESULTS AND DISCUSSIONS

From the implementation of the custom utility-preserving algorithm for maintaining the utility and privacy, we obtained a perturbed dataset Z with alpha = 0.6 and epsilon 0.1. After obtaining the perturbed dataset Z with the privacy and utility loss functions and gradient descent optimization algorithm, Assessment of perturbation operation 'PO' (privacy loss + Utility loss functions) on the said dataset. Twenty six algorithms were used for this purpose. The evaluation of the performance was done with the help of Lazy predict library. In this process, we select machine learning algorithms that are applicable to the perturbation process on datasets D' or Z. Next, we evaluated the impact of 'PO' (privacy loss + Utility loss functions) on the performance of the machine learning algorithms on the perturbed dataset D' or Z using various metrics. These metrics = {m1, m2, m3} , where m1 is accuracy, m2 is f score and m3 = execution time of the algorithm to finally predict the target 'y'. It is desired that the performance of the machine learning algorithms must fall to prove the effectiveness of the 'PO'. Hence, to prove this, following attack scenarios were simulated to match real life cases reported in incidence reports of privacy breach and loss

of trust. Hence, this section demonstrates a simulation of an attack scenario, which involves predicting the user's location based on their activities and possible geographic area in which they might be . To simulate this attack scenario, it is assumed that the attacker will train the possible case using a machine learning model, specifically a Classifier function, to predict the user location from the activities and possible geographic coordinates. The table [2] gives detailed information on the possible ways in which an attack may be conducted by an attacker.

Table 2. Possible methods with which the attacker may use to Predict Information

S.No	**Possible Attack Scenarios:**	**Possible Machine Learning Technique**
1	Predicting the duration of the activity based on the latitude and longitude coordinates. X =['lat','lon','location','activity'] y = 'user id' , where X are the inputs and y is the target that attack was to predict	Regression [15] /Classification
2	Predicting the start time or end time of the activity based on the latitude and longitude coordinates. X =['lat','lon','activity','user id'] y = 'start_time' | 'end_time' , where X are the inputs and y is the target that the attack was to predict.	Regression
3	Predicting the user who performed the activity based on the latitude and longitude coordinates. X = ['lat','lon','activity','user id'] y =['user id'] | ['user_name'] , where X are the inputs and y is the target that the attack was to predict.	Classification
4	Predicting the distance travelled during the activity based on the latitude and longitude coordinates. [activity] -> [[lat]/[lon]] = D , where D is the distance travelled.	Regression, calculus
5	Predict the location of the user based on their activity data (latitude and longitude coordinates). X = ['lat','lon','activity','user id'] ,y =['location'] , where X are inputs and y is target variable.	Regression/Classification
6	Predicting the user who performed the activity based on the latitude and longitude coordinates.	Classification

	X = ['lat','lon','activity'] ,y =['user_id']| ['user_name'] , where X are inputs and y is target variable.	
7	Predicting the user who performed the activity based on the latitude and longitude coordinates, activity, and location name. X = ['lat','lon','activity','location'] ,y =['user_id']| ['user_name'] , where X are inputs and y is target variable.	Classification
8	Predicting the activity performed based on the latitude and longitude coordinates and time. X = ['start_time','end_time','lat','lon'] ,y =['activity'], where X are inputs and y is target variable.	Classification
9	Predicting the velocity of movement based on the latitude and longitude coordinates & time**.** [activity] ->[[lat]/[lon]] = D/[start_time,end_time] , where V is the velocity	Regression, Calculus

As per the problem formulated and suggested solution here we explain the final outcome of this research work. Therefore to evaluate the integrity of the perturbed dataset Z we determine if any sensitive information can still be inferred from the perturbed data. In other words, we want to check and ensure that the perturbation process did not result in unintended disclosures of sensitive data to the attacker.

Evaluating the Impact of PO :

Since, the dataset contains information about 'n' number of users over 'dn' days, with each user performing between 1 and 6 activities [walk =1 , running =2, driving =3 , sitting =4, shopping =5, sleeping =6] per day. The resulting dataset, df, contains the columns user id, day, activity, location (locations = ['park' = 1 ,'mall' =2 , 'office' =3 , 'home' =4]), lat, and lon. Using this dataset, there are nine possible ways with which an attacker may try to make an educated guess on the location of the User 'U'.

X = ['lat','lon','activity','location_name,user_name']
y =['location_name']
Where X are inputs and y is target variable.

To evaluate the effectiveness of the perturbation operation 'PO' in reducing accuracy, f-score and time , additional calculations need to be performed using the given metrics. The number of machine learning algorithms impacted by 'PO' can be determined by counting the algorithms for which the accuracy, recall, or precision decreased after perturbation. Similarly, the effectiveness ratio is calculated as the ratio of the number of algorithms impacted by 'PO' to the total number of algorithms used. However , due to multiple

metrics it is hard to interpret the outcome. Hence, a metric known as NPI (Normalized Performance Index) was computed and to know the difference between the original dataset D and Perturbed Data 'Z' difference percentage was computed. These calculations can be observed from table 3.

Table 3. Evaluating the impact of PO on D→Z

S.No	**Original Data 'D' Baseline Performance of Machine Learning Models**				**Perturbed Data 'Z' Performance of Machine Learning Models**				**Difference Percentage = ((Perturbed NPI - Original NPI) / Original NPI) * 100**
S.No	**O_m1**	**O_m2**	**O_m3**	**O_NPI**	**Z_m1**	**Z_m2**	**Z_m3**	**Z_NPI**	**DP**
LSVC	0.9812	0.98111	0.33841	0.916561	0.91722	0.9455	0.030552	0.837037	-96.6666703
RC	0.975	0.993137	0.03681	0.886622	0.91166	0.9112	1.200414	0.940397	35.39184282
LDA	0.9668	0.883465	1.39228	0.984348	0.93472	0.914	0.423104	0.877342	-57.01680555
RCCV	0.9715	0.882449	0.44779	0.892414	0.83055	0.8191	0.221556	0.766216	-75.17339772
LR	0.8662	0.874613	0.29486	0.81159	0.82916	0.828373	0.557179	0.801726	-31.34722352
DTC	0.755	0.758791	0.67008	0.747645	0.71777	0.926268	0.363971	0.74494	-51.31769035
BC	0.8432	0.732393	0.42627	0.768265	0.81805	0.792707	0.02904	0.731546	-96.22005379
P	0.8563	0.776732	0.03497	0.750297	0.81083	0.72039	0.014472	0.704062	-98.07116279

RFC	0.8596	0.725823	0.017437	0.735251	0.8125	0.713268	0.405324	0.742013	-44.87267334
ETC	0.4514	0.83311	0.48812	0.569585	0.41388	0.813621	0.333708	0.525785	-41.41208073
GNB	0.8548	0.838469	0.39984	0.804405	0.81138	0.868389	0.01763	0.749108	-97.80831713
DC	0.8403	0.840106	0.021227	0.758335	0.81278	0.802699	0.032457	0.731722	-95.71996263
ETC	0.741947	0.749172	0.039015	0.673821	0.71444	0.761685	0.014705	0.658642	-97.81767065
SGD	0.731702	0.821747	0.01771	0.687316	0.81444	0.861685	0.019058	0.749078	-97.22718638
NSVC	0.849175	0.809404	0.022943	0.754621	0.81444	0.861685	0.168458	0.764018	-77.67646121
CCCV	0.247079	0.195015	0.202933	0.227045	0.202778	0.301758	0.132931	0.225487	-41.45174617
SVC	0.743905	0.729084	0.159953	0.681064	0.7875	0.759211	0.011461	0.701409	-98.31719069
NC	0.80693	0.870677	0.013779	0.746739	0.891667	0.859809	0.02445	0.795388	-96.72576362
PAC	0.630032	0.722173	0.029407	0.597612	0.791667	0.759767	0.039958	0.706926	-93.31371971
LGBMC	0.833438	0.785231	0.047953	0.740427	0.793056	0.779844	0.210448	0.730832	-71.577497
QDA	0.540164	0.692763	0.253543	0.557282	0.690278	0.661923	0.021251	0.614869	-96.18666757

LP	0.711442	0.680649	0.025575	0.633617	0.602778	0.663468	0.08281	0.568988	-86.93059881
BNB	0.729007	0.769714	0.099604	0.678279	0.686111	0.66103	0.009361	0.610912	-98.61988905
LS	0.712116	0.746926	0.011265	0.652474	0.683333	0.66823	0.575116	0.66798	-11.8560911
KNC	0.708613	0.602078	0.693477	0.675139	0.783333	0.753104	0.021077	0.698039	-96.87812389
BC	0.61418	0.63299	0.025439	0.560949	0.572222	0.526263	0.015034	0.502716	-97.31989848

Table 4. List of Abbreviation

Abbreviation	Definition
LSVC	Linear SVC
RC	Ridge Classifier
LDA	Linear Discriminant Analysis
RCCV	Ridge Classifier CV
LR	Logistic Regression
DTC	Decision Tree Classifier
BC	Bagging Classifier
P	Perceptron
RFC	Random Forest Classifier
ETC	Extra Trees Classifier
GNB	Gaussian NB
DC	Dummy Classifier
ETC	Extra Tree Classifier
SGD	SGD Classifier
NSVC	NuSVC
CCCV	Calibrated Classifier CV
SVC	SVC
NC	Nearest Centroid
PAC	Passive Aggressive Classifier
LGBMC	LGBM Classifier
QDA	Quadratic Discriminant Analysis
LP	Label Propagation
BNB	Bernoulli NB
LS	Label Spreading
KNC	KNeighborsClassifier

ABC	Ada Boost Classifier
E	Entropy
H	Homogeneity
R	Regularity
S	Skewness
K	Kurtosis
P	Proximity to successive steps
pn	Proneness of attack or "attack susceptibility"
D	Original location dataset
Z	Perturbed dataset
X	Random variable representing the location of the target individual
Y	Random variable representing the attributes of the target individual at their location
alpha	Balancing parameter to find the trade-off between privacy and utility
epsilon	Privacy budget
grad X	Gradient descent optimization
PO	Perturbation operation
m1, m2, m3	Metrics used for evaluation (accuracy, f score, execution time)
o_m1	Accuracy of original dataset
o_m2	F-score of original dataset
o_m3	Execution time of original dataset
oNPI	Original Normalised Performance Index
z_m1	Accuracy of Perturbed dataset
z_m2	F-score of Perturbed dataset
z_m3	Execution time of Perturbed dataset
zNPI	Perturbed Normalised Performance Index
DP	Difference Percentage

Upon examination, it is evident that the percentage difference in most instances is negative, indicating a decline in the performance of the Machine Learning algorithms (M*) in 25 cases. To facilitate a more straightforward interpretation, a metric known as the Effectiveness Ratio (E.R) has been computed. This metric gauges the effectiveness of the Privacy Operation (PO) in terms of percentage and is defined as follows:

Effectiveness Ratio (E.R) = *(Number of algorithms impacted by 'PO') / (Total number of algorithms)*

As per the data in the table, it is clear that the number of impacted algorithms is 25. These impacted algorithms have Negative Performance Impact (NPI) values of: [0.009361, 0.011461, 0.014472, 0.014705,

0.01763, 0.015034, 0.019058, 0.021077, 0.02445, 0.030552, 0.02904, 0.021251, 0.032457, 0.039958, 0.08281, 0.168458, 0.221556, 0.210448, 0.423104, 0.363971, 0.405324, 0.132931, 0.333708, 0.557179, 0.575116]. Furthermore, the corresponding serial numbers of these matching values are [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25].
It is noteworthy that only one algorithm remained unaffected according to these results, with an NPI value of [1.200414].

Effectiveness Ratio (E.R): Based on the above numerical results, it can be confidently asserted that the process of modifying the properties of the original dataset was successful, with an effectiveness ratio of 0.9615384615384616.

VII. CONCLUSION AND FUTURE DIRECTIONS

This research has made significant strides in addressing the privacy concerns associated with location-based datasets. A comprehensive review of the existing literature revealed that these datasets are particularly vulnerable to homogeneity attacks and other similar threats. The inherent numerical properties of location-based datasets were identified as a significant challenge, as they can be easily exploited by attackers using machine learning and other advanced methodologies. To address these challenges, a custom function was developed to strike a balance between preserving the utility of the dataset and ensuring the privacy of the data. This function was designed to transform the original dataset in a way that reduces its vulnerability to potential attacks, while still retaining its utility for legitimate purposes. The successful implementation of this function resulted in a transformed dataset that effectively mitigated the risks identified in the original data. The effectiveness of this approach was evaluated using a suite of 26 distinct machine learning algorithms. These algorithms were chosen due to their observed performance degradation when applied to the transformed dataset. The primary objective of these experiments was to ascertain the effectiveness of the perturbation operation (PO) approach. The results of these experiments provided valuable insights into the potential of PO in enhancing the privacy of location-based applications. Furthermore, the custom function developed in this research relies on the addition of Laplacian noise to the original data. While this approach is effective in perturbing the data, it may also reduce the utility of the dataset. Future research could explore alternative methods of perturbation that maintain the utility of the data while enhancing privacy. In conclusion, this research has made significant contributions to the field of privacy preservation in location-based datasets. However, further research is needed to address the challenges identified in this study and to further enhance the effectiveness of the PO approach.

REFERENCES

1. D. Srivastava, H. Pandey, and A. K. Agarwal, "Complex predictive analysis for health care: A comprehensive review," *Bull. Elect. Eng. Inform.*, vol. 12, no. 1, pp. 521–531, Feb. 2023.
2. K. Chin. "Biggest data breaches in US history [updated 2023] | upguard." Third-Party Risk and Attack Surface Management Software.
3. J. Colnago, L. Cranor, and A. Acquisti, "Is there a reverse privacy paradox? An exploratory analysis of gaps between privacy perspectives and privacy-seeking behaviours," *Proc. Privacy Enhancing Technology.*, vol. 2023, no. 1, pp. 455–476, Jan. 2023. Accessed: Mar. 8, 2023.

4. L. L. Dhirani, N. Mukhtiar, B. S. Chowdhry, and T. Newe, "Ethical dilemmas and privacy issues in emerging technologies: A review," *Sensors*, vol. 23, no. 3, p. 1151, Jan. 2023. Accessed: Mar. 8, 2023.
5. Y. Ren *et al.*, "DistPreserv: Maintaining user distribution for privacy-preserving location-based services," *IEEE Trans. Mobile Comput.*, p. 1, 2022. Accessed: Mar. 8, 2023
6. J. Zhang, Q. Huang, Y. Huang, Q. Ding, and P.-W. Tsai, "DP-TrajGAN: A privacy-aware trajectory generation model with differential privacy," *Future Gener. Computer. Syst.*, vol. 142, pp. 25–40, May 2023.
7. J. Zhao, J. Mei, S. Matwin, Y. Su, and Y. Yang, "Risk-Aware individual trajectory data publishing with differential privacy," *IEEE Access*, p. 1, 2021. Accessed: Mar. 8, 2023.
8. X. Zhang, F. Breitinger, E. Luechinger, and S. O'Shaughnessy, "Android application forensics: A survey of obfuscation, obfuscation detection and deobfuscation techniques and their impact on investigations," *Forensic Sci. International: Digit. Investigation*, vol. 39, p. 301285, Dec. 2021.
9. P. Ram Mohan Rao, S. Murali Krishna, and A. P. Siva Kumar, "Privacy preservation techniques in big data analytics: A survey," *J. Big Data*, vol. 5, no. 1, Sep. 2018. Accessed: Mar. 8, 2023.
10. A. Majeed and S. Lee, "Anonymization techniques for privacy preserving data publishing: A comprehensive survey," *IEEE Access*, p. 1, 2020. Accessed: Mar. 8, 2023
11. Z. Zheng, Z. Li, J. Li, H. Jiang, T. Li, and B. Guo, "Utility-aware and privacy-preserving trajectory synthesis model that resists social relationship privacy attacks," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 3, pp. 1–28, Jun. 2022.
12. P. Arora. "Data breach incidents: Why reporting matters and how to do it." Medium.
13. Y. Zhao and J. Chen, "A survey on differential privacy for unstructured data content," *ACM Comput. Surv.*, Jan. 2022. Accessed: Mar. 8, 2023.
14. D. M. Bittner, A. E. Brito, M. Ghassemi, S. Rane, A. D. Sarwate, and R. N. Wright, "Understanding privacy-utility tradeoffs in differentially private online active learning," *J. Privacy Confidentiality*, vol. 10, no. 2, Jun. 2020.
15. J. Wu and F. Zhang, "Privacy-Preserving regression modelling and attack analysis in sensor network," in *Cloud Computing and Big Data*. Cham: Springer International Publishing, 2015, pp. 354–366.
16. A. Zigomitros, F. Casino, A. Solanas, and C. Patsakis, "A survey on privacy properties for data publishing of relational data," *IEEE Access*, vol. 8, pp. 51071–51099, 2020.
17. D. Parmar and U. P. Rao, "Towards privacy-preserving dummy generation in location-based services," *Procedia Comput. Sci.*, vol. 171, pp. 1323–1326, 2020.
18. H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–36, Jan. 2021.
19. B. Niu, X. Zhu, H. Chi, and H. Li, "Pseudo-Location Updating System for privacy-preserving location-based services," *China Commun.*, vol. 10, no. 9, pp. 1–12, Sep. 2013. Accessed: Mar. 8, 2023.
20. Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, and G. Xu, "Constructing dummy query sequences to protect location privacy and query privacy in location-based services," *World Wide Web*, Jul. 2020. Accessed: Mar. 8, 2023.
21. Gupta, R., & Rao, U. P. (2022). Investigating and devising privacy preserving approaches for location-based services. In *Intelligent Technologies: Concepts, Applications, and Future Directions* (pp. 129-148). Singapore: Springer Nature Singapore.
22. R. Khan *et al.*, "Θ-Sensitive k-Anonymity: An Anonymization Model for IoT based Electronic Health Records," *Electronics*, vol. 9, no. 5, p. 716, Apr. 2020.

23. Kaur, G., & Gupta, R. (2021, October). A study on location based services and TTP based privacy preserving techniques. In *2021 International Conference on Advances in Computing and Communications (ICACC)* (pp. 1-5). IEEE.
24. T. Patel and V. Patel, "Data privacy in construction industry by privacy-preserving data mining (PPDM) approach," *Asian J. Civil Eng.*, vol. 21, no. 3, pp. 505–515, Feb. 2020.
25. L. U. Khan, W. Saad, Z. Han, and C. S. Hong, "Dispersed federated learning: Vision, taxonomy, and future directions," *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 192–198, Oct. 2021.
26. A. Zigomitros, F. Casino, A. Solanas, and C. Patsakis, "A survey on privacy properties for data publishing of relational data," *IEEE Access*, vol. 8, pp. 51071–51099, 2020.
27. Gupta, R., & Rao, U. P. (2017). Achieving location privacy through CAST in location based services. *Journal of Communications and Networks*, *19*(3), 239-249.
28. Q. Zhang, X. Zhang, M. Wang, and X. Li, "DPLQ: Location-based service privacy protection scheme based on differential privacy," *IET Inf. Secur.*, Aug. 2021.
29. Ramprashath R, Dr.R.Nallakumar, Nagajothi Karthiga.N, Murugeswari. P Personality Divination with Profile Analysis and Questionnaire Screening Using Machine Learning doi: 10.31838/ecb/2023.12.s3.1062023.19/04/2023
30. Kaur, G., Gupta, R., Hooda, N., & Gupta, N. R. (2022). Machine learning techniques and breast cancer prediction: A review. *Wireless Personal Communications*, *125*(3), 2537-2564.
31. V. Thammanna Gowda, "Stack and deal: An efficient algorithm for privacy preserving data publishing," in *2nd Int. Conf. Mach. Learn. &Trends (MLT 2021)*. AIRCC Publishing Corporation, 2021.
32. Y. Saleem, M. H. Rehmani, N. Crespi, and R. Minerva, "Parking recommender system privacy preservation through anonymization and differential privacy," *Eng. Rep.*, Oct. 2020. K. M. Chong, "Privacy-preserving healthcare informatics: A review," *ITM Web Conf.*, vol. 36, p. 04005, 2021.